Data Driven Update of Load Forecasts in Smart Power Systems using Fuzzy Fusion of Learning GPs

Miltiadis Alamaniotis
Electrical & Computer Engineering
University of Texas at San Antonio
San Antonio, TX, USA
miltos.alamaniotis@utsa.edu

Antonio Martinez-Molina
Dept. Architecture
University of Texas at San Antonio
San Antonio, TX, USA
Antonio.martinez-molina@utsa.edu

Georgios Karagiannis
Dept. of Mathematical Sciences
Durham University
Durham, UK
georgios.karagiannis@durham.ac.uk

***Abstract*— One of the pillars in developing smart power systems is the use of load forecasting methods. In particular load forecasting accommodates decision making pertained to the operation of power market. In this paper, a new method for real-time updating very short-term load forecasting is proposed. The goal of the method is to accurately predict the load demand value in the next 5 minutes and accordingly update the daily forecast. To that end, the proposed method implements an ensemble of homogeneous learning Gaussian processes which are trained on slightly different training datasets. The predicted values are then fused using a fuzzy inference system in order to obtain a single value which is used to correct the precomputed forecast. The proposed method is tested on a set of real-world data taken from a major US area and is benchmarked against the naïve forecasting method. Results highlight the superiority of our method against the benchmarked method exhibiting an increase in forecasted accuracy by 50% in most cases.**

***Index Terms*—Load forecasting, forecast update, Gaussian process, fuzzy inference, smart power.**

I. Introduction

Advances in machine learning and computing technologies over the last two decades have made possible the modernization of power infrastructure. In a more visionary approach, the power grid is a fully intelligent and autonomous mega system that makes decisions over its operational needs [1]. This vision is known as the vision of smart power systems, in which information flows both ways: from generation centers to consumers and vice versa [2].

One of the pillars in building smart power systems is load forecasting [3]. The utilization and storage of information available in the power system allows intelligent tools to perform data driven load forecasting. In particular, machine learning tools may learn observed load demand values and then make predictions over the future demand. In a pure data driven forecasting the tools utilize only data patterns and do not take into consideration any predetermined model. The advantage of the data driven forecasting is its flexibility, scalability and ease of implementation, while it can be updated at low computational cost [4].

There are several types of load forecasting with respect to forecasting horizon. In particular, very short-term load forecasting (VSTLF) refers to forecasts from a few minutes up to a few hours, short term (STLF) from a few hours up to a week, medium term (MTLF) covers a week up to a year while long term (LTLF) refers to horizons longer than a year [5]. Smart power systems may utilize all the above types of forecasting but fundamental to its implementation are the STLF and VSTLF, both used for operational and market-based decisions [4].

Electricity load forecasting has been a topic of interest in the research community for long time. Several methods have been proposed using a variety tools and approaches, with the vast majority of them based upon data-driven tools taken from the statistics and the machine learning domains [5] either for VSTL [6-11] or STLF [12-16]. However, there is a limited number of works on combining those two types of load forecasting and none of them take into consideration the updates of forecasts in the context of smart power systems and price directed markets.

In this work a new method for updating load forecasts is proposed. The method utilizes a set of learning Gaussian processes (GPs) [17] that are trained on different datasets. The training sets consist of the most recent observed load demand values and are utilized for forecasting of the next ones. The individual GP forecasts are subsequently fused to obtain a single value that is subsequently used to correct the daily forecast for that hour. Notably, the forecasts fusion is conducted by a simple fuzzy logic inference system [18]. The underlying idea of the proposed method is the the use of several data driven GPs as the mean to capturing of the most recent load dynamics via data assimilation. Therefore, the contribution of the paper lies in the following:

- A novel data driven Fuzzy-GP load update method,
- The application of the method for first time in the updating of 5 min ahead load forecasts.

The rest of the paper is organized as follows. Section II briefly describes the basics of learning GP, while section III

978-1-6654-3597-0/21/$31.00 ©2021 IEEE

presents the load update method. Further, section IV provides and discusses the results obtained on a set of real-world datasets. Lastly, section V concludes the paper and highlights its findings.

II. Learning Gaussian Processes

One of the preeminent classes of tools in the area of machine learning is the class of kernel machines. As kernel machines are identified those models that may be expressed as a function of a kernel. A kernel is any valid analytical function that is comprised of a basis function f(x) and takes the following form:

$$k(x_1, x_2) = f(x_1)^T f(x_2) \tag{1}$$

which is also known as the kernel trick [17].

A Gaussian process is any set of random variables that share a joint probability distribution that takes a Gaussian form. More particularly a GP is expressed as:

$$GP \sim N(m(x), C(x', x)) \tag{2}$$

where $m(x)$ is the mean function and $C(x', x)$ is the covariance function of the process.

In order to derive the learning GP framework the following two choices are made: i) the mean function is taken to be equal to 0 (convenient choice), and ii) the covariance function is set equal to a kernel function [17], i.e., $C(x',x) = k(x',x)$. Notably, the selection of the form of the kernel function determines the output of the GP, a property that is useful to several applications.

GP are considered as a learning method given that they require to under training process. By representing the population of training datapoints as N, which is comprised of N pairs of (x,t) (known outputs **t** for known inputs **x**), then a new input is taken as x_{N+1} and its (unknown) target as t_{N+1}. In this GP framework, the joint distribution between the N training datasets and the input x_{N+1} takes a Gaussian form. Based on the above assumptions, the GP framework provides a predictive distribution of Gaussian form [20]:

$$m(\mathbf{x}_{N+1}) = \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{t}_N \tag{3}$$

$$\sigma^2(\mathbf{x}_{N+1}) = k - \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{k} \tag{4}$$

where the mean and variance are a function of the kernel **k**. In Eq. (3) and (4), $\mathbf{C}_N$ is covariance of the training dataset, **k** is a vector of covariance values of the the input x_{N+1} and training data, and k is the kernel output of $k(x_{N+1}, x_{N+1})$.

III. Load Update Method

In this section the proposed load forecast updating method is presented. The goal of the method is to create an intelligent ensemble of GPs that have been trained on different training sets. The underlying idea is that through training of multiple GPs the system will be able to capture the most recent load dynamics, and subsequently provide more accurate very short-term forecasts by updating the current ones. The block diagram of the proposed method is depicted in Fig. 1 where all steps are provided.

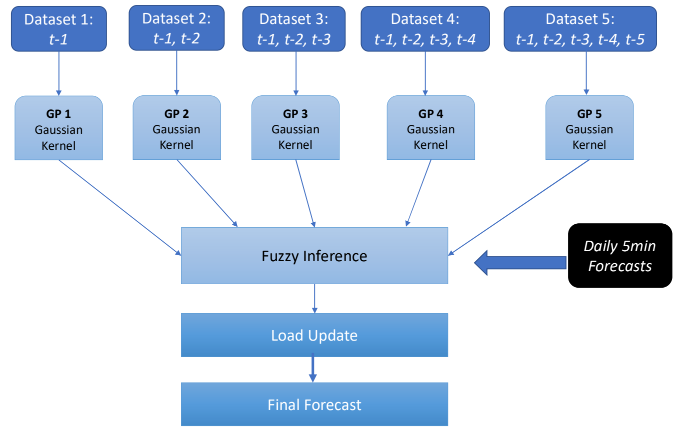

Figure 1. Update Load forecasts method.

Initially, a group of 5 datasets is defined where each dataset encompasses a different number of recent measurements. Getting into details, the first dataset is comprised of the most recent measurement (*t-1* in Fig. 1), the second datasets of the 2 most recent, the third dataset from the 3 most recent, the fourth from the 4 most recent and the fifth from the 5 most recent measurements respectively. Therefore, we created a diverse group of datasets of different length expecting that we possibly capture the latest load dynamics. The different lengths are selected because we are not aware the exact point when the dynamics may initiate.

In the next step, five GP models are adopted where all of them are equipped with a Gaussian kernel function whose analytical form is given below:

$$k(x_1, x_2) = \left(2^{1-\theta_1}/\Gamma(\theta_1)\right)\left[\sqrt{2\theta_1}\,|x_1 - x_2|/\theta_2\right]^{\theta_1} K_{\theta_1}\left(\sqrt{2\theta_1}\,|x_1 - x_2|/\theta_2\right) \tag{5}$$

where there are two hyperparameters θ_1, θ_2 with θ_1 taken equal to 3/and the other one is determined in the training process. Each of the GP models is exposed to a single dataset in order to train its hyperparameter and then use it for obtaining the predictive distribution from Eq. (3) and (4).

Once the models are exposed to datasets and trained, then they are utilized for making next term prediction and more particularly of the next 5 min forecast. Hence, a set of 5 forecasts are obtained and forwarded to a fuzzy inference system (FIS). In addition to the GP forecasts, the FIS also utilizes the daily forecasts that are usually obtained the previous day. The aim of the FIS is to merge the individual forecasts with the previous forecasts and update the load forecasts. The underlying idea is that by capturing the recent dynamics we may be able to correct the error introduced by the day ahead forecasting. For this reason, the FIS weights the GP accordingly and fuses them with the daily forecast to provide a new updated forecast. The block diagram showing the steps of the of FIS forecast fusion operation is depicted in Fig. 2.

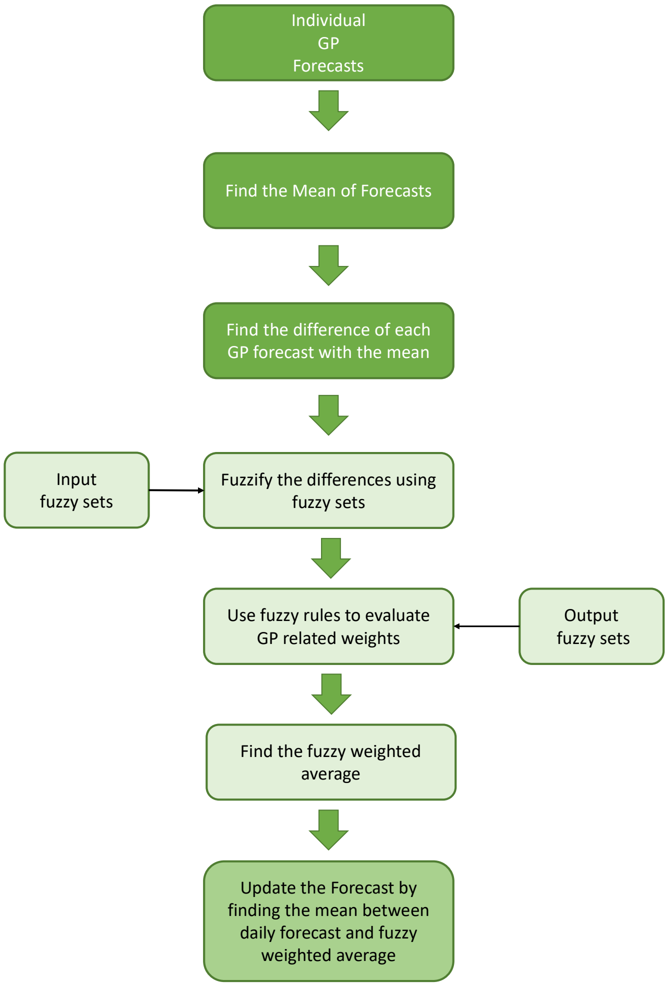

Figure 2. Block diagram of fuzzy inference system process.

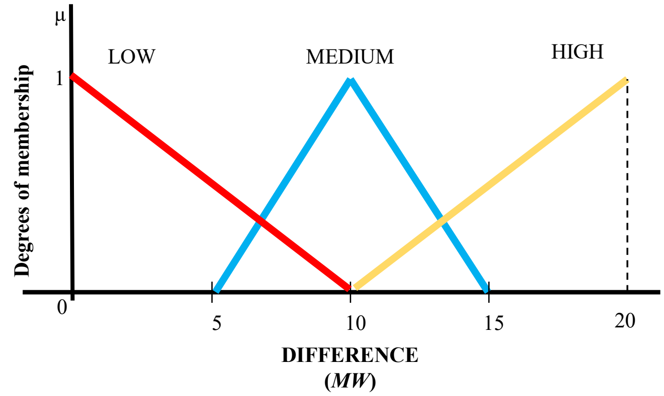

Figure 3. Input fuzzy sets for fuzzifying the variable D_i in Eq. (6).

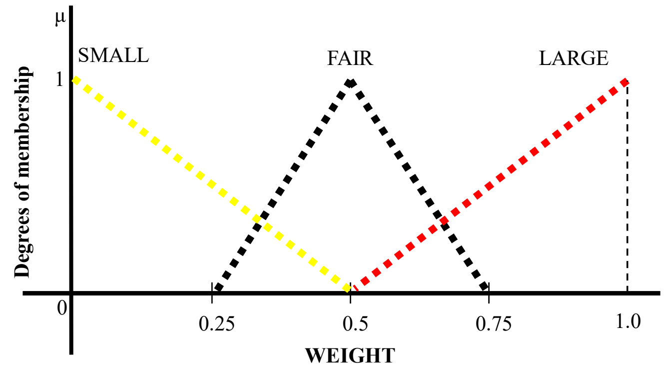

Figure 4. Output fuzzy sets for wieght values.

As we observe in Fig. 2, initially we compute the mean value of the five GP forecasts denoted as *M*. Then, the differences between the mean and each of the GP models are found as given below:

$$D_i = |\, GP_i - M \,|, \quad i=1,..,5 \tag{6}$$

where GPi denotes the forecast of i^{th} GP model. In the next step, the computed differences are fuzzified using the input sets that are depicted in Fig. 3. The input fuzzy sets span the range of [0 20] *MW* where we implicitly take into consideration the forecasts that are +/- 20 MW from the mean value. This range is adopted empirically [21-22] and is a way to detect and reject the outliers: forecasts outside of this range are not taken into consideration for the load update. The fuzzy sets (i.e., input sets) used for fuzzifying the differences in Eq. (6) are presented in Fig. 4.

Once the difference values are fuzzified, then they are forwarded to the rule inference module that is implemented as a set of fuzzy rules with the aid of a group of fuzzy output sets. The output sets, which are given in Fig. 5, span the range [0 1] that is used to evaluate a group of weight parameters.

Both input and output fuzzy sets are put together to form a group of IF/THEN rules that consist the core of the FIS. The rules defined in this work are given below:

- If DIFFERENCE is LOW, Then WEIGHT is LARGE,
- If DIFFERENCE is MEDIUM, Then WEIGHT is FAIR,
- If DIFFERENCE is HIGH, Then WIEGHT is SMALL,

where we observe that there are three rules that connect the difference with the weight variable. The above rules are used to evaluate the weight values. In our work, we assign a single weight to each GP forecast and as a result the number of weight parameters is equal to 5 – matches the number of GP models-. The underlying idea is that the smaller the difference is, the higher the weight we assign to the respective GP model. Following that evaluation strategy we subsidize those forecasts that are closer to their mean in the hope that the mean is very close to the actual demand. At this point, it should be stated that the output of the fuzzy rules is a fuzzy set that undergoes defuzzification in order to obtain a single value as the final output of FIS. The defuzzification method adopted in this work is the centroid as defined below [18]:

$$y = \frac{\sum_{i=1}^{N} y_i \mu_{out}(y_i)}{\sum_{i=1}^{N} \mu_{out}(y_i)} \quad (7)$$

where y_i being the elements of the fuzzy set, and N denotes the element population in the fuzzy set.

Once the weights are evaluated then they are put together to get a new weighted average:

$$Wa = [w_1 GP_1 + w_2 GP_2 + w_3 GP_3 + w_4 GP_4 + w_5 GP_5] / 5 \quad (8)$$

where the weights w_i are taken as the output of Eq. (7). In the last step the weighted average is further fused with the initial load forecast by utilizing the following formula:

$$U(t) = (Wa(t) + Load(t)) / 2 \quad (9)$$

where U(t), Wa and Load are the updated load, weighted average and the initial load forecast at time t. To make it clearer, Eq. (9) is the average value of the weighted average and the initial load.

Overall, it is the value U(t) that is taken as the updated load forecast, and replaces the initial Load(t) value.

IV. Results

A. Test Setup

The proposed method is tested on a set of real-world data taken from the hub of Connecticut New England ISO [23]. The data are in the form of 5 min load demand and measured in MW units. In this work, we have selected data taken from the year 2019 and contains the 5 min demand of a whole week in February, i.e., specifically the week of Feb 2-8, and the special days (holidays in US) of New Year Day and Martin Luther King day.

The daily forecast method that is obtained is the naïve approach, where the real values of the previous day are used as the forecasts of the next. The results are recorded in terms of the mean average percentage error (MAPE) per day [22] for both the naïve forecasting approach and our method.

B. Test Results

In this section, we record and presents the results obtained by applying our method and the naïve forecaster to the test datasets.

The results obtained for the week February 2-8, 2019 are presented in Fig. 5. There, we provide the MAPE taken with the naïve forecaster and the respective MAPE taken by applying the proposed forecast updating method. In addition, the Fig. shows the improvement in terms of percentage in forecasting attained by applying the GP-fuzzy load update. We observe that the presented method succeeds in improving the load forecasts as shown for all the seven tested days.

In particular, the improvement for all days except for February 6 is about 50%, while for February 5 is about 10%. The average improvement in load forecasting for the tested days is equal to 2.7%. The obtained results confirm our hypothesis that the data driven part of the method that uses GP is able to capture the load dynamics and significantly improve the load forecasts in very short term horizons.

For visualization purposes, we provide the updated forecasted load against the actual load demand in Fig. 6-8 for the days of February 2, 4 and 8. The plotted curves exhibit the ability of the method to capture the trend of the real demand curve.

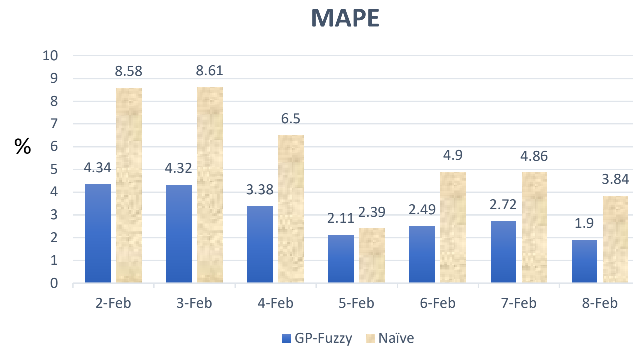

Figure 5. Output MAPE results for the week of February 2-8, 2019.

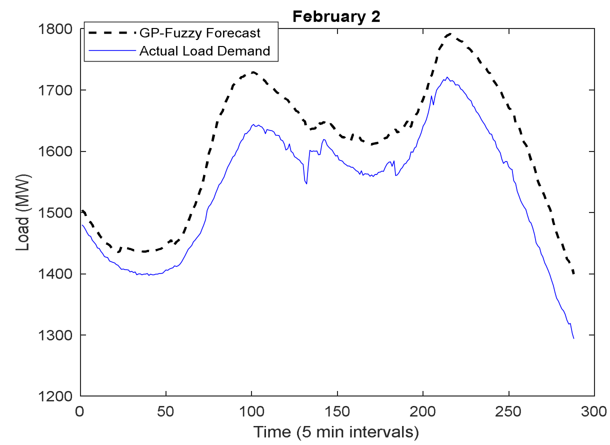

Figure 6. Updated forecasts against actual demand for February 2, 2019.

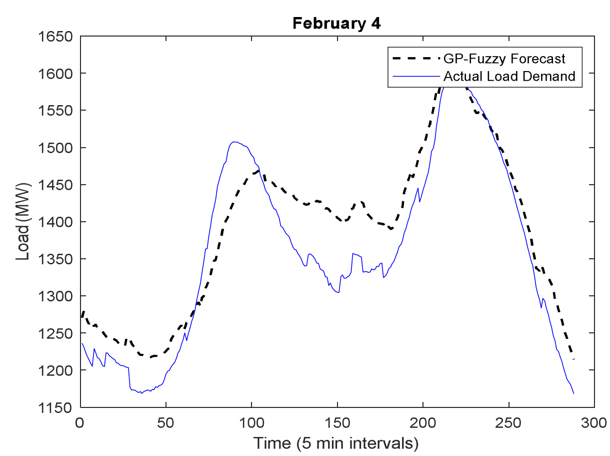

Figure 7. Updated forecasts against actual demand for February 4, 2019.

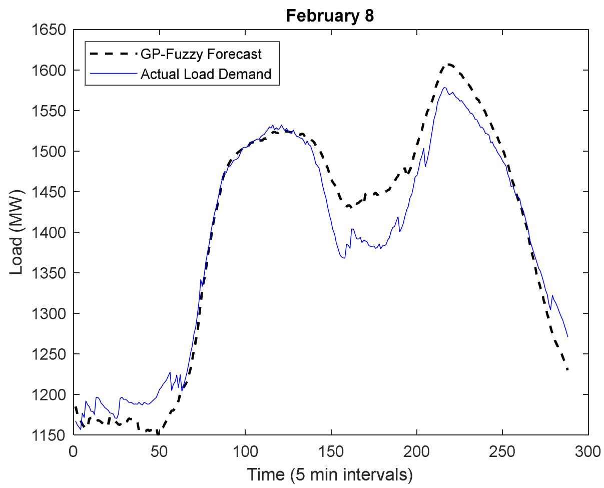

Figure 8. Updated forecasts against actual demand for February 8, 2019.

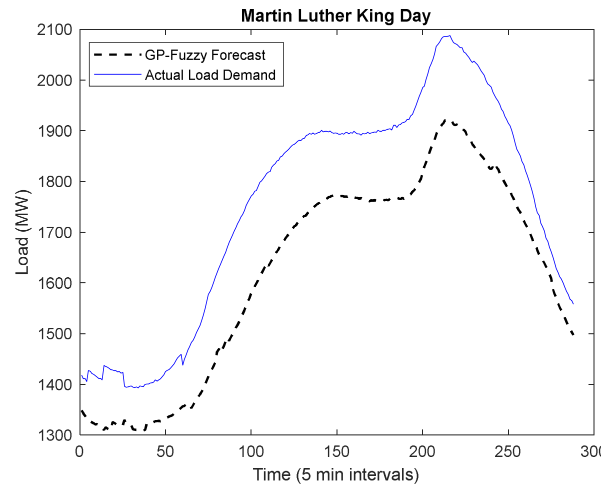

Figure 11. Updated forecasts against actual demand for Martin Luther King Day.

Additionally, we test our method on a set of special days. The recorded results for the tw special days are given in Fig. 9, while we visualize the forecasts taken from the presented method against the actual demand in Fig. 10-11.

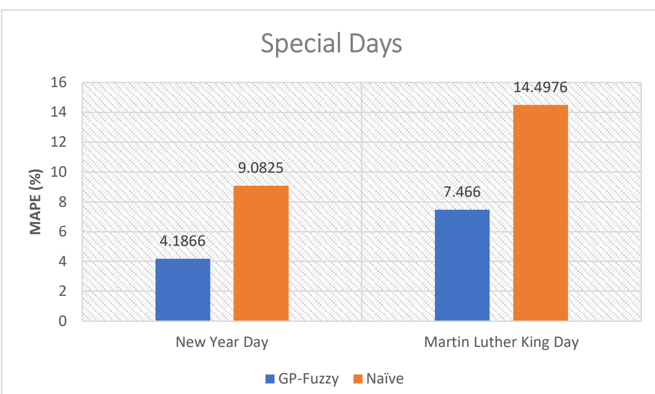

Figure 9. Output MAPE results for the tested special days.

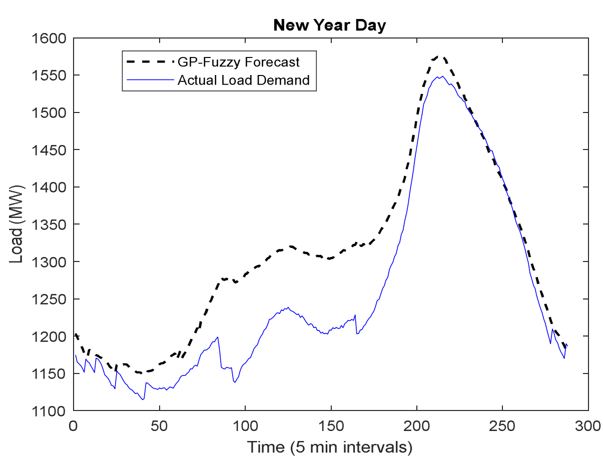

Figure 10. Updated forecasts against actual demand for New Year Day.

Similar to the tested week of February, the results for the two special days also exhibit an improvement in the accuracy of forecasting by using the data-driven update method presented in this paper. More specifically, we observe that for the New Year Day, our method provided an improvement of about 5% - it reduced the initial forecast MAPE from 9% to 4% (approximate numbers). Likewise, for the Martin Luther King Day there is also improvement which is about 7% as can be inferred from Fig. 9. Furthermore, we can safely conclude that the presented method managed to capture the special days load dynamics and improve the forecasts in very short terms. This improvement is also apparent from the load curved in Fig. 10 and 11 for the two special days respectively.

V. Conclusion

In this paper, a new method for updating load forecasting was presented. The method is suited for smart power systems where demand forecasting is essential for the overall operational decision making of the system. In particular, we presented a data driven method that utilizes two tools taken from the AI library: kernel based Gaussian processes and fuzzy inference systems.

The aforementioned tools are integrated in such way so that we implement a new method for updating the daily ahead forecast of demand. The proposed method utilizes the most recent measurement in order to provide an update forecast over the next load demand value. Its testing on a set of real world 5 min load data obtained from New England ISO, exhibit significant improvement on the initial forecast. Results showed an improved accuracy in terms of MAPE for all tested cases – a whole in February and two special days-.

In the future, we intend to extensively test the method in a larger variety of data (from other seasons as well), and check different type of kernel function for the GP models. In addition, the effect of fuzzy resolution (i.e., the number of fuzzy sets) will be also examined in order to infer whether a higher number of input and output fuzzy sets will further improve the forecasting accuracy.

References

[1] F. Li, W. Qiao, H. Sun, H. Wan, J. Wang, Y. Xia, Z. Xu, and P. Zhang, 2010. "Smart transmission grid: Vision and framework," *IEEE Trans. on Smart Grid*, vol. 1(2), pp.168-177, 2010.

[2] Alamaniotis, M. and Tsoukalas, L.H., 2017, September. Utilization of virtual buffer in local area grids for electricity storage in smart power systems. In Proc. *2017 IEEE North American Power Symposium (NAPS)*, pp. 1-6.

[3] L.H. Tsoukalas, and R. Gao, "Inventing energy internet The role of anticipation in human-centered energy distribution and utilization," In *Proc. 2008 SICE Annual Conference*, pp. 399-403.

[4] C. Ye, Y. Ding, P. Wang, and Z. Lin, "A data-driven bottom-up approach for spatial and temporal electric load forecasting," *IEEE Transactions on Power Systems*, *34*(3), pp. 1966-1979, 2019.

[5] M. Alamaniotis, and L.H. Tsoukalas, "Implementing smart energy systems: Integrating load and price forecasting for single parameter based demand response," In *Proc. 2016 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pp. 1-6.

[6] M. Q. Raza and A. Khosravi, "A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings," *Renewable and Sustainable Energy Reviews*, vol. 50, pp. 1352–1372, 2015.

[7] W. Charytoniuk and M.-S. Chen, "Very short-term load forecasting using artificial neural networks," *IEEE Transactions on Power Systems*, vol. 15, no. 1, pp. 263–268, 2000.

[8] C. Guan, P. B. Luh, L. D. Michel, Y. Wang, and P. B. Friedland, "Very Short-Term Load Forecasting: Wavelet Neural Networks With Data Pre-Filtering," *IEEE Transactions on Power Systems*, vol. 28, no. 1, pp. 30–41, 2013.

[9] M. Alamaniotis, A. Ikonomopoulos, and L. H. Tsoukalas, "Evolutionary Multiobjective Optimization of Kernel-Based Very-Short-Term Load Forecasting," *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1477–1484, 2012.

[10] M. Alamaniotis, "Synergism of Deep Neural Network and ELM for Smart Very-Short-Term Load Forecasting," *2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*, 2019.

[11] C. Guan, P. B. Luh, L. D. Michel, and Z. Chi, "Hybrid Kalman Filters for Very Short-Term Load Forecasting and Prediction Interval Estimation," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 3806–3817, 2013.

[12] N. Mohan, K.P. Soman, and S.S. Kumar, "A data-driven strategy for short-term electric load forecasting using dynamic mode decomposition model," *Applied energy*, *232*, pp. 229-244, 2018.

[13] M. Alamaniotis, and L.H. Tsoukalas, "Multi-kernel assimilation for prediction intervals in nodal short term load forecasting," In *2017 19th IEEE International Conference on Intelligent System Application to Power Systems (ISAP)*, pp. 1-6.

[14] M. Alamaniotis, A. Ikonomopoulos, and L.H. Tsoukalas, "A Pareto optimization approach of a Gaussian process ensemble for short-term load forecasting," In *Proc. 2011 16th IEEE International Conference on Intelligent System Applications to Power Systems*, pp. 1-6.

[15] S.N. Fallah, M. Ganjkhani, S. Shamshirband, and K.W. Chau, "Computational intelligence on short-term load forecasting: A methodological overview," *Energies*, *12*(3), pp. 393(1-21), 2019.

[16] A. Baliyan, K. Gaurav, and S.K. Mishra, "A review of short term load forecasting using artificial neural network models," *Procedia Computer Science*, *48*, pp. 121-125, 2015.

[17] C.E. Rasmussen, and C. Williams, Gaussian processes for machine learning, MIT Press: Boston, 2006.

[18] L.H. Tsoukalas and R.E. Uhrig, Fuzzy and Neural Approaches in Engineering, New York: Wiley Int, 1997.

[19] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211-244, 2001.

[20] C. M. Bishop, *Pattern Recognition and Machine Learning*, New York: Springer, 2006.

[21] M. Alamaniotis, and L.H. Tsoukalas, "Multi-kernel assimilation for prediction intervals in nodal short term load forecasting," In *2017 19th IEEE International Conference on Intelligent System Application to Power Systems (ISAP)*, pp. 1-6.

[22] M. Alamaniotis, M., A. Ikonomopoulos, and L.H. Tsoukalas, "Evolutionary multiobjective optimization of kernel-based very-short-term load forecasting," *IEEE Transactions on Power Systems*, vol. 27(3), pp. 1477-1484, 2012.

[23] New England ISO, www.iso-ne.com